

NOTES ON MATCHING

JONATHAN HIRATA

1. INTRODUCTION AND DEFINITIONS

This paper assumes basic knowledge of definitions and concepts as they pertain to graph theory. With that in mind, let's begin with the main topic of these notes: matching. For now we will start with general definitions of matching. Later we will look at matching in bipartite graphs then Hall's Marriage Theorem.

1.1. General Definitions.

Definition 1.1. A **matching** of graph G is a subgraph of G such that every edge shares no vertex with any other edge. That is, each vertex in matching M has degree one.

Definition 1.2. The **size** of a matching is the number of edges in that matching.

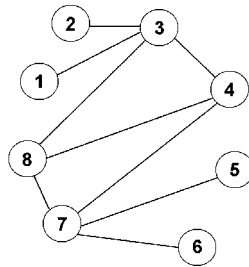


FIGURE 1

Consider the graph in Figure 1. Denote the edge that connects vertices i and j as (i, j) . Note that $\{(3, 8)\}$ is a matching. Obviously we can get more. The pairs $\{(3, 8), (4, 7)\}$ also make a matching. That is a matching of size two. Can we get a matching of size three? Yup, it's $\{(2, 3), (4, 8), (5, 7)\}$. Can we do even better? Well, a matching of size four means that every vertex is paired, but vertices 1 & 2 must both be paired with vertex 3. So no, three is the best we can do. We call it a *maximum* matching.

Definition 1.3. A matching is **maximum** when it has the largest possible size. Note that for a given graph G , there may be several maximum matchings.

Definition 1.4. The **matching number** of a graph is the size of a maximum matching of that graph.

Thus the matching number of the graph in Figure 1 is three.

Definition 1.5. A matching of a graph G is **complete** if it contains all of G 's vertices. Sometimes this is also called a **perfect** matching.

Thus no complete matching exists for Figure 1.

1.2. Matching in Bipartite Graphs. Let's begin with a recap of what a bipartite graph is:

Definition 1.6. A **bipartite graph** is a graph whose vertices can be divided into two disjoint sets such that no edge connects two vertices of the same set.

It is common to use the terms *left* and *right* to describe the two sets of vertices. A *balanced* bipartite graph is one that has an equal number of left and right vertices.

Definition 1.7. Consider a subset $S \subseteq L$ of left vertices of bipartite graph G . Let n be the number of right vertices the set S is connected to. Then the **deficiency** D of set S is defined as:

$$D(S) = \begin{cases} |S| - n & \text{if positive} \\ 0 & \text{otherwise} \end{cases}$$

Define the **left deficiency** D_L of a bipartite graph as the maximum such $D(S)$ taken from all possible subsets S . **Right deficiency** D_R is similarly defined.

As an example, let's consider the complete bipartite graph $K_{3,2}$. Recall that $K_{m,n}$ stands for a complete bipartite graph with m left vertices and n right vertices.

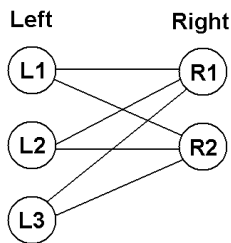


FIGURE 2. $K_{3,2}$

If we consider the subset $S = \{L_1, L_2\}$, we see that the deficiency of this subset is zero, as the number of neighbors of S is at least the size of S . But if we consider the subset $S = \{L_1, L_2, L_3\}$, we find a deficiency of one, as these three left vertices connect to only two right vertices. Since there exists no larger deficiency, the *left deficiency* of $K_{3,2}$ has a value of one. Now let's look at the right side of our graph. Consider the subset $S = \{R_1\}$. Since this vertex connects to 3 left vertices, this subset has a deficiency of zero. In fact it is easy to see that the *right deficiency* of our graph is also zero.

With these concepts in place, we will now offer a proposition, to be proven later.

Proposition 1.8. The matching number of a bipartite graph G is equal to $|L| - D_L(G)$, where L is the set of left vertices. Likewise the matching number is also equal to $|R| - D_R(G)$, where R is the set of right vertices.

Referring back to Figure 2, we see that $|L| - D_L(G) = |R| - D_R(G) = 2$. And clearly a matching of size 2 is the maximum matching we are going to find. We will now switch gears slightly and focus on a particular subcase of the above proposition. We will now focus on the case where we can find a complete matching. The result we are after is known as Hall's Marriage Theorem.

1.3. Hall's Marriage Theorem. Philip Hall in 1935 gave us the condition for when a complete matching is possible in a bipartite graph. An easy way to visualize this is to consider the following situation: Suppose we are pairing up N boys and N girls (if they were not both N then clearly there is no way for a matching of our bipartite graph to be complete). Now each girl comes up with a list of acceptable mates that she likes, some subset of the N boys. Since these boys are of the gentlemanly type, none of them will reject a proposal if given to them. This situation can be represented by a bipartite graph, where an edge represents the event that a specific girl likes a specific guy. One such possible arrangement is given in Figure 3.

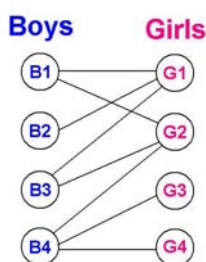


FIGURE 3

We can now state Hall's marriage condition:

Definition 1.9. *Hall's marriage condition holds when every subset of r girls likes at least r boys. This is exactly the same as saying Hall's marriage condition holds when $D_L(G) = 0$ (or $D_R(G) = 0$) for a balanced bipartite graph G .*

An equivalent condition can be created by interchanging 'boy' and 'girl' in the definition above. Also note that the marriage condition only applies to the case where we have an equal number of boys and girls, i.e. when we have a balanced bipartite graph. Can you see how you would relate this condition to a bipartite graph? Here is the main theorem of this section.

Theorem 1.10 (Hall's Marriage Theorem). *Hall's marriage condition is both necessary and sufficient for the existence of a complete match in a bipartite graph. That is to say, iff Hall's marriage condition holds for a bipartite graph, then a complete matching exists for that graph.*

Looking at Figure 3 we can see that this graph does not meet the marriage condition. If we take the set of girls $\{G_3, G_4\}$, they are both paired with only one boy, thus they have a deficiency of one. This violates the marriage condition. Here it is easy to see that no complete matching exists because of this: only at most one of $\{G_3, G_4\}$ could get married, the other one won't be. If we look at the marriage condition from the boy's side, we also see a violation of the marriage condition. If we look at the set of boys $\{B_1, B_2, B_3\}$, we see that they collectively like the same two girls. This set also has a deficiency of one. So a similar problem arises: at most two boys from our set can possibly get married; the third one is doomed to be a bachelor forever.

Notice that the marriage theorem is a subcase of Proposition 1.8, where $N = |L| = |R|$ and $D_L(G) = 0$. Both theorems say that in this case, the matching number is N , or in other words there exists a complete matching.

It is easy to see that the marriage condition is *necessary* for a complete matching, but Hall's marriage theorem asserts that it is also *sufficient*.

Proof of Hall's Marriage Theorem. Since necessity is easy to see, we need to prove that the marriage condition is also sufficient. That is to say, if the marriage condition holds, then there exists a complete matching.

We will use induction to prove our desired result. Given a balanced bipartite graph G , assume that the marriage condition holds. We will induct on r , the size of a subset S of left vertices of our graph G . What we need to do is show that if the marriage theorem holds for r , then it also holds for $r + 1$, and eventually the entire graph G . But first the base case, where $r = 1$.

If $r = 1$ and the marriage condition holds true for any subset of size one, then clearly it can be paired with a right vertex. Then we have a complete matching for any one vertex. Base case is done.

Now for the inductive step. Assume that for *any* r -sized subset S , the marriage condition holds and so does the marriage theorem. Now we need to show, based on the previous assumptions, that the marriage theorem also holds true for $r + 1$. Let the set of vertices that S connects to be denoted as S' . Now consider any $(r + 1)$ th left vertex, v_{r+1} . Now we split the problem into three cases.

Case 1. The vertex v_{r+1} is connected to some vertex not in S' . Then clearly there exists a complete matching among these $(r + 1)$ vertices. This is illustrated in Figure 4.

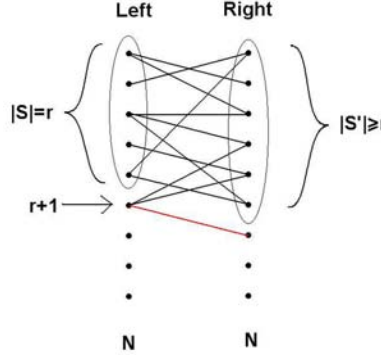


FIGURE 4. Case 1

Case 2. The vertex v_{r+1} is only connected to vertices in S' , and $|S'| = |S|$. In this case, it is clear that this violates the marriage condition which we have assumed to be true for our entire graph G . So this case is impossible.

Case 3. The vertex v_{r+1} is only connected to vertices in S' , and $|S'| > |S|$. Then we use our assumption that for *every* r -sized subset we can find a complete matching. In particular, we look at every r -sized subset that includes $(r - 1)$ members of S plus vertex v_{r+1} . Now what we must show is that given this property (that all these r -sized subsets have a complete matching), this implies that we can

create an $(r + 1)$ -sized matching. So WLOG, let's remove vertex v_1 and consider all the other vertices (including vertex v_{r+1}) as our r -sized subset. Let's call this subset S_r . Now consider all complete matchings for S_r . Let S'_r be the subset of S' that is in every complete matching of S_r . If vertex v_1 is connected to only points in S'_r , then this contradicts the marriage condition which we have assumed to be true. Thus v_1 is connected to some point not in S'_r , thus a complete matching exists for all $r+1$ points. This is illustrated in Figure 5. The bold edges represent one possible complete matching for vertices v_2, \dots, v_{r+1} . Because of our original assumptions, the red edge is guaranteed to exist, thus establishing our inductive step.

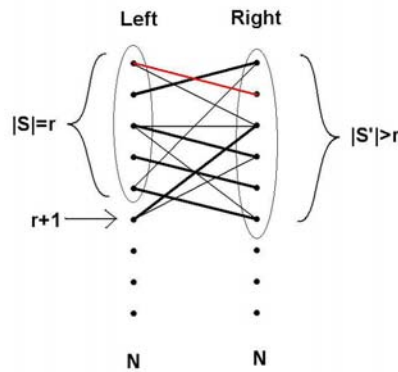


FIGURE 5. Case 3. The red edge is guaranteed to exist.

With the completion of our inductive step, it can be seen that the marriage theorem is true for our entire graph G . And since the size of G was not stipulated, the marriage theorem thus holds true for any such arbitrarily large graph.

◇

Proof of Proposition 1.8. Now we will offer a hand-wavy proof of the proposition above. It is basically exactly the same as the proof of Hall's Marriage Theorem, except now we have a left deficiency $D_L > 0$. And instead of a complete matching, we have a matching of size $|L| - D_L$. You will see that if you follow the same inductive argument, the desired result follows.

◇

We will now describe an algorithm for find a maximum matching given a bipartite graph G . The main step in this algorithm is called **augmenting**, which involves taking a matching of size n , and making it $n + 1$. To begin this algorithm, start with the first vertex in L , the set of left vertices in G . Go ahead and match it to some vertex in R (if such an edge doesn't exist, just skip it for now). Now look at the next vertex in L , and if possible, match it to an unpaired vertex in R . If you can't, just skip that vertex for now. Continue this process for all of L . Here, the augmenting step involves matching two unpaired vertices that share an edge. Once you do this for all vertices in L , this does not guarantee that we have found a maximum matching. We must now try to match even more unpaired vertices. But obviously they cannot immediately be paired with each other. What we are looking for now is a path from an unpaired L vertex to an unpaired R vertex that alternates between matched and unmatched edges.

Definition 1.11. An **alternating path** is a path that alternates between matching and non-matching edges.

Definition 1.12. An **augmenting path** is an alternating path that starts and ends on unmatched vertices.

An example of such a situation is in Figure 6.

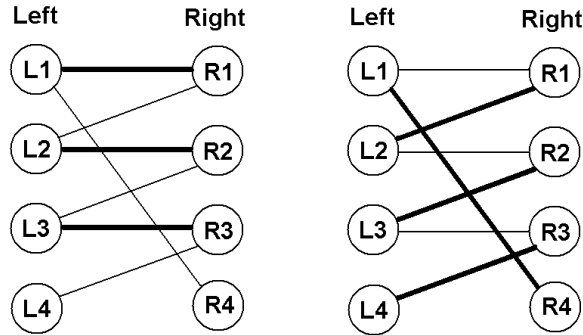


FIGURE 6. Augmentation: Before and After

In the left half of Figure 6, we have gone down through the entirety of L , but clearly an augmenting path exists between vertices L_4 and R_4 . Now by breaking old matchings and adding new ones, we can augment our matching into a matching that is larger by one. Basically what we do is to reverse the state of all the edges in our path, i.e. change unmatching edges into matching ones and vice-versa. And although we won't prove it, it is guaranteed that through this algorithm we will find a maximum matching.

Before we move on, I'd like to propose the following lemma:

Lemma 1.13. Consider a bipartite graph G for which the marriage condition holds. Now take any pair of vertices $\{v_1, v_2\} \in G$ that are connected through an edge. If we allow for multiple edges, the operation of adding another edge between those two vertices preserves the marriage condition in graph G . Likewise, removing an edge from between two vertices where a multiple edge already exists also preserves the marriage condition.

Proof. By adding an edge to two points where an edge already existed (say $\{v_1, v_2\}$), that changes nothing in terms of the number of vertices connected to any subset containing a subset of vertices containing vertex v_1 (or v_2). So the marriage condition must still hold. The same argument is used when removing an edge, as long as at least one edge still remains between those two vertices. \diamond

Intuitively, it doesn't matter if a girl makes 15 proposals to the same guy, thus creating 15 edges between them. Just the fact that the girl likes him creates an edge, which is all we need. So 15 edges still function the same as if there were only one edge.

We can now give an example of Hall's Marriage Theorem at work.

Example 1.14. Take a regular deck of 52 playing cards and randomly deal them into 13 piles of 4 cards each. Then by the marriage theorem, there exists a way to pull out one card from each pile so that you have one card from every rank (ace, 2, ..., king).

Proof. To prove the example above, we must first represent the problem as a bipartite graph. Then we must show that the marriage condition holds for our graph. After we have done that, we can use Hall's Marriage Theorem to prove the example.

So the first step is to model this problem as a bipartite graph. How? Have each of the 13 piles represent a vertex on the left, and each of the 13 ranks represent a vertex on the right. An edge exists between two vertices iff that pile contains that rank. For example, if an ace is found in pile number 1, we draw an edge between the top two vertices. The keen observer will note the possibility of multiple edges. Multiple edges will arise when two cards of the same rank (say two kings) appear in the same pile. So this is how we model our problem as a bipartite graph.

The next step is to show that the marriage condition holds in our graph. Notice that each vertex on the left (the piles) have degree 4 (because they have 4 cards in them), and every vertex on the right (the ranks of the cards) also have degree 4 (because each of the 4 cards in each rank must be placed in a pile). Now we have a specific case of one of your homework problems.

With the proof of said homework problem, the marriage condition is shown to hold true, and by Hall's Marriage Theorem, there exists a complete matching in our bipartite graph. Applying this property back to our piles of cards, we see that it is possible to pull one of each rank from the 13 piles, according to the edges in our complete matching that we have shown must exist. \diamond

2. STABLE MARRIAGE PROBLEM

The problem can be stated as such: Suppose you are a match-maker and have a group of N men and N women. Now each of these N men and women have ranked (after a long and arduous process) the members of the opposite gender according to his or her own preference. Your job as match-maker is to come up with a matching M , pairing up every man and woman to make N couples. Let us define a **rogue couple** in a matching M as a man and a woman such that they prefer each other to the person they are currently paired with in M . Clearly, rogue couples are something that you should avoid in your final decision as match-maker, as these rogue couples are likely to elope and deprive you of your commission.

Definition 2.1. A **stable matching** is a matching that has no rogue couples.

The next logical question is does there always exist a stable matching in such a situation? The answer is yes, and the result is known as the Stable Marriage Theorem. The following is an algorithm by Dale Gale and Lloyd Shapley in 1962 to find such a stable matching. The algorithm is sorta like actual human behavior.

- (1) In the beginning of each round, every man not engaged proposes to his favorite girl not yet crossed off his preference list.
- (2) Then each woman looks at her new proposals and also the man she is engaged to (if she has one) and then picks her favorite from the group. They become engaged while all the other men get permanently rejected from her.
- (3) All rejected men cross off their lists the woman who rejected them.

The cycle repeats until there is a round when no man is rejected, i.e. everyone is paired. This algorithm guarantees that everyone gets married, and that the matching is stable.

Theorem 2.2. *Everybody gets married.*

Proof. First note that once a woman is proposed to, she is engaged to be married and will never lose that. She can only trade up to better suiters on her list. And since each man was ranked every woman, the only way that a man will not be married is if he is rejected by every woman. Assume there exists a man that is not married, then there also must be a woman who is unmarried. But this unmarried woman must have been proposed to at one point by the unmarried man, thus she must be married. Contradiction, so everybody gets married. (Yay). \diamond

Theorem 2.3. *The Gale-Shapley algorithm produces a stable matching.*

Proof. Assume the opposite, that the matching is not stable. So there must be a rogue couple, $B - G$. Suppose B married G' (who he likes less than G) and that G married B' (who she likes less than B). That means B must have proposed to G at one point and she rejected him. But the only way that G would reject B is if she had a proposal from somebody she liked better. But women can only trade up from round to round, which means she must like B' better than B . Contradiction, so the matching must be stable. (Double yay). \diamond

Theorem 2.4. *The algorithm terminates in at most $N(N - 1) + 1$ rounds.*

Proof. This just puts an upper-bound on the number of rounds that can take place in this algorithm. Notice that at every round except the last round, a man crosses a woman off his list. And since every man gets married, he can cross off no more than $(N - 1)$ women. So there are no more than $N(N - 1)$ cross-offs and thus no more than $N(N - 1) + 1$ rounds. \diamond

Who do you think fares better in this algorithm? The men or the women? The proposers or the acceptors? One may think that since the men get rejected, and the women do all the accepting, that the women do better. But on the other hand, the men start at the top (of their respective lists) and work their way down, while the women start with their worst suitor and work their way up. Plus it isn't so clear what the meaning of "faring better" exactly is. The following formalizes the question and gives a more rigorous result.

Let S be the set of all stable matchings. We know from Theorem 2.2 that $S \neq \emptyset$. We define for each person P his or her **realm of possibility** as $\{Q \mid \{P, Q\} \in M \in S\}$. That is, Q is within the realm of possibility for P iff there exists a stable matching where P marries Q .

Definition 2.5. *A person's **optimal mate** is that person's favorite from the realm of possibility.*

Definition 2.6. *A person's **pessimal mate** is that person's least favorite from the realm of possibility.*

Now for a pair of shocking results:

Shocking Result 2.7. *The algorithm pairs every man with his optimal mate!*

Shocking Result 2.8. *The algorithm pairs every woman with her pessimal mate!*

Proof of Shocking Result 2.7. Assume that there exists a boy that does not get his optimal girl in the Gale-Shapley algorithm. Let boy B be the first boy in time (resolving ties arbitrarily) that gets rejected by his optimal mate, call her G . Define Z to be the boy who caused G to reject B . Clearly G prefers Z over B . Since B was the first boy rejected by his optimal mate, boy Z has not (yet) been rejected by his optimal mate. So that means, from the eyes of Z , girl G is his optimal girl or is better than his optimal girl. Now let's look at a stable matching $M \in S$ where B marries his optimal girl G . This is guaranteed to exist by definition. Also note that M was not found by the Gale-Shapley algorithm. In this matching, boy Z is married to some other girl Y . But here the rogue pair $Z - G$ becomes apparent as Z prefers G over Y . That is because Z would prefer G over anybody else in a stable matching. And in the other half of the rogue pair, girl G ! prefers Z over B (who she is currently married to in matching M) by definition. So we do indeed have a rogue pair and thus M is not a stable matching, which of course is a contradiction. Hence the Gale-Shapley algorithm pairs every man with his optimal mate. \diamond

Proof of Shocking Result 2.8. We can use Shocking Result 2.7 to help us with this proof. Let's look at the matching M produced by the Gale-Shapley algorithm. Pick any girl G married to a boy B in matching M . Now consider any other stable matching S such that G is no longer paired to B . Because M was created by the Gale-Shapley algorithm, B was paired to his optimal mate G and thus prefers her to whoever he is currently paired with in S . But because S is stable, girl G cannot prefer B over her mate in S (or else we would have a rogue pair). Thus any girl would prefer any other stable match to her mate under the Gale-Shapley algorithm. \diamond

So it turns out that the proposers fare much better than the acceptors. I guess it pays to be aggressive in courtship, as long as you don't mind a few rejections at first.

2.1. Applications. This sort of marriage problem arises in many applications in the real world. Probably the most famous example is in matching new MDs to hospitals, known as the National Resident Matching Program (NRMP). Each fourth year med student fills out a list of his/her top twenty hospitals, and the hospitals do the same for their choice in doctors. Then the algorithm is run with the hospitals acting as the boys and the new residents as girls. But with a slight variation that there are now many girls to each boy. The doctors find out their assignments on *match day*, which is a huge event. But of course every hospital ends up getting their optimal choice and the doctors end up with their pessimal choice.

Not surprisingly, the Gale-Shapley algorithm is also used by at least one large dating agency.

MIT professor and co-founder of Akamai Tom Leighton reports that Akamai uses a variation of the Gale-Shapley algorithm to match web traffic to servers. [1][2] Before, Akamai used combinatorial algorithms to optimize the matching, but this got to be too slow as the number of servers and traffic increased. So they then switched to the Gale-Shapley algorithm, which is much faster. In this case, the web traffic corresponds to the boys and the web servers to the girls. The servers have preferences based on latency and packet loss; the traffic has preferences based on the cost of bandwidth.

2.2. Variations on the Problem. But of course the situation we have described rarely occurs exactly as such. We have already looked at two examples. In the hospital/resident problem (also known as the college admissions problem [6]) every boy is allowed more than one girl. Of course the opposite may be applied, that every girl is allowed many guys.

When considering the problem of pairing men and women, more obvious discrepancies arise. What if there aren't exactly N men and N women? And more realistically, each person (man and woman) is likely to have ranked only some of the members of the opposite sex, creating a partial preference list. And furthermore, sometimes a pairing may be considered unacceptable to either the boy or the girl, or both. When such situations arise, it is possible that some people will go unmarried under the Gale-Shapley algorithm. Just consider the possibility that a woman thinks none of her possible suitors are good enough for her. She will go unmarried. But for those who do get married, the matching is a stable one.

Another variation is if we allow same-sex marriages, that is, all participants belong to a single pool. This is known as the stable roommates problem [6]. This is because any potential roommate can be paired with any other potential roommate. So we no longer have two groups where a pair must consist of one member from each group, but one giant group where all pairings are possible.

It is interesting to note that in the stable roommates problem, it is possible that no stable matching exists. The idea is to create a preference triangle among three of the participants, with a fourth person who is everybody's last choice. See Figure 7.

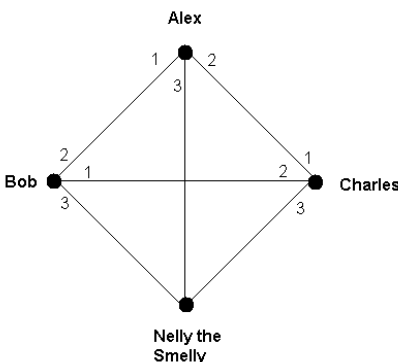


FIGURE 7. Nobody wants a smelly roommate

It turns out that Nelly's preferences don't even matter (not to mention that also any preference list Nelly can come up with would have a certain symmetry with his other possible preference lists).

Theorem 2.9. *The situation in Figure 7 has no stable matching.*

Proof. Without loss of generality (because all matchings are symmetric), we can arbitrarily pair Alex-Bob and Charles-Nelly. But then Bob would rather be paired with Charles, and Charles would rather be paired with anybody other than Nelly. Thus Bob-Charles is a rogue pair and the matching is unstable. And by symmetry, all matchings are unstable. \diamond

3. EULERIAN PATHS

We begin this section with a definition:

Definition 3.1. An **Eulerian path** is a path on a graph that visits each edge exactly once.

An example of an Eulerian path is the following, which many of you may recognize as a classic puzzle from childhood: Draw the following pictures without lifting up your pencil.

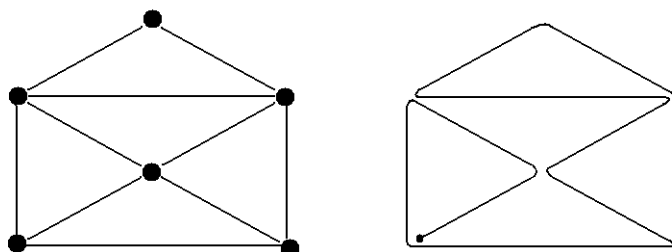


FIGURE 8. An Eulerian Path

Note that although it is not drawn this way, Eulerian paths are allowed to cross itself at vertices. Perhaps it is possible to prove that any Eulerian path can be drawn without crossing itself, but that is getting off topic. Also note that Eulerian paths are not necessarily unique for a given graph.

The origins of this topic spawn from the famous Seven Bridges of Königsberg problem, which Euler tackled in 1736. The problem asks, can the graph in Figure 9 be traversed without crossing the same edge twice?

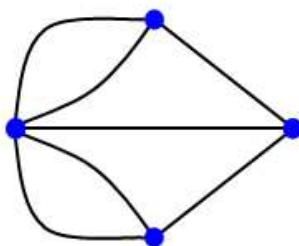


FIGURE 9. Representation of the Königsberg bridges

It turns out the answer is no. And the determining property was noted by Euler:

Theorem 3.2. A graph G has an Eulerian path iff G is connected and has at most two vertices of odd degree.

Proof. Note that Eulerian paths must start and stop at vertices of odd degree (if they exist in graph G). If not, then there must be some edge not traversed. Now consider a graph G that has more than two vertices of odd degree. Then it must

have more than two starting and stopping points, which is impossible. This proves the \Rightarrow direction.

Now for the opposite direction, consider a graph G that has at most two vertices of odd degree. Here we will describe an algorithm (due to Fleury, 1883) as well as the proof. Start at an odd vertex in G (if it exists). Otherwise, start at any vertex. Now at each step, traverse an edge and delete it, as long as deleting that edge does not make G disconnected. Such a move is guaranteed to exist (can you figure out why?). Thus after each step we've reduced the problem by one edge, and since there are finitely many edges, eventually we will run out. The path that we took is an Eulerian path. \diamond

Returning to the Königsberg bridges in Figure 9, we can see that there are four vertices with odd degree, and thus by the above theorem, has no Eulerian path.

One last puzzle involves Figure 10. Without lifting your pencil, can you cross every line *segment* exactly once?

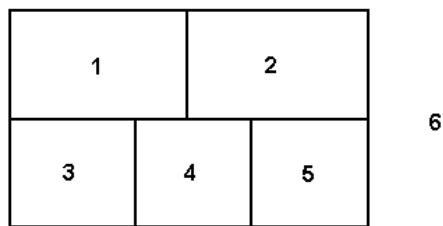


FIGURE 10

To answer this problem, we must represent each room (including the outside) as a vertex and each line segment as a graph edge. This is known as its **dual graph**. This is represented in Figure 11.

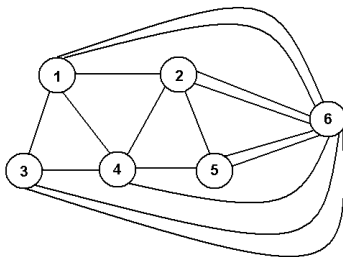


FIGURE 11

Now the question is something we are familiar with, namely, does there exist an Eulerian path in Figure 11? The answer is no, because vertices 1, 2, 4, and 6 all have odd degree.

4. LIST COLORINGS

Let's first review basic concepts in coloring.

Definition 4.1. A **proper coloring** is a coloring of each vertex in a graph such that no edge is connecting vertices of the same color.

Definition 4.2. The **chromatic number** $\chi(G)$ of a graph G is the least number of colors needed for the existence of a proper coloring of G . Sometimes this is plainly called the **coloring number**.

As an example, all bipartite graphs have a chromatic number of two. Just color all the L vertices one color and all the R vertices another color.

Definition 4.3. A **list coloring** is a choice function where each vertex is assigned a color from a prescribed list for that vertex.

Definition 4.4. The **list coloring number** $ch(G)$ of graph G is the smallest integer k such that for any possible list configuration there exists a proper list coloring for G where k is the size of the lists for each vertex.

To help further clarify the concept of list coloring number, say we are given a graph G , and we know its list coloring number is $ch(G)$. Then we give our graph G along with $ch(G)$ to an adversary. This adversary then comes up with a list for each vertex in G where each list contains $ch(G)$ different colors. Now no matter how our adversary concocts the lists, we are guaranteed to find a proper matching of G . This is the essence of the list coloring number.

Now the definition of list coloring number says nothing about how many colors there are *globally* in the entire graph G . For instance, if every list of size k had completely different colors, we would have $k|V|$ colors total, where $|V|$ is the number of vertices in G . But if our adversary did that he would not be a very smart adversary indeed, because then we could find a proper list coloring by choosing any color from each vertex's list. So intuitively, for the adversary to make it as hard as possible for us to find a proper list coloring, he would choose as few colors globally as he could. Or in other words, every list for each vertex should consist just of the same k colors. But this would turn out to be completely wrong. Consider the following (partial) example in Figure 12.

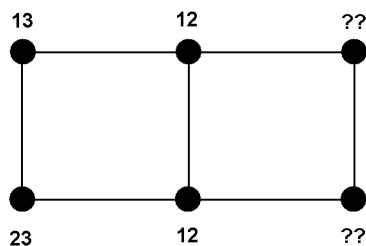


FIGURE 12

Here we are determining if this graph is 2-list colorable. Consider the top middle vertex. If we choose the color 1 for its color, we then get a contradiction. That's

because the bottom middle vertex must then be 2, and the top left vertex must be 3. But then the bottom left vertex has no options left. So this is an example of how using more than 2 colors globally can give us a contradiction. Can you label the other two vertices so we also get a contradiction if we label the top middle vertex 2 rather than 1? Doing so would thus show that the graph is not 2-list colorable. Notice that the chromatic number of Figure 12 is two. Indeed, it is in fact a bipartite graph, which becomes more visible when we swap the positions of the middle two vertices.

5. HOMEWORK EXERCISES

1. Prove: If all vertices of graph G have the same degree and G is bipartite, then there exists a complete matching in G . Furthermore, the edges of G can be partitioned into complete matchings.

2. For this question you must know the definition of latin square:

Definition 5.1. A **latin square** is an N by N array of integers where each element is from the set $\{1, \dots, N\}$ and in no row or column does the same integer appear more than once.

Notice that all (correctly) completed Sudoku puzzles form a latin square. But Sudoku puzzles have one extra constraint involving 3×3 blocks.

Prove: If you have a k by N array of integers from 1 to N obeying the same condition (with $1 \leq k < N$), then you can complete the missing rows to make a latin square.

Hint: Induct on k by showing you can always get one more row that obeys the latin square condition. Do so by turning the problem into a bipartite graph with N vertices on the left (representing each of the N columns) and N vertices on the right (representing the numbers 1 through N). An edge exists between two vertices if that number is eligible to be placed in that column.

3. a) Find a graph (on 6 vertices) whose coloring (chromatic) number is 2 but whose list coloring number is 3. How many vertices do you need to make the list coloring number of a bipartite graph equal to 4?

b) Show how to find a matching in a bipartite graph having N vertices in each part in which each vertex has degree d obeying $d = 2^k$ by wandering over a total of $2Nd$ edges. (Hint find cycles and modify them).

REFERENCES

- [1] Leighton, Tom & Rubinfeld, Ronitt, "6.042 Lecture Notes: Graph Theory II," <http://www.cs.princeton.edu/courses/archive/fall106/cos341/handouts/graph2.pdf>
- [2] Meyer, Alber R. "6.042 Lecture Notes: Stable Marriage," <http://courses.csail.mit.edu/6.042/spring07/ln5.pdf>
- [3] Wikipedia, "Eulerian Path," http://en.wikipedia.org/wiki/Eulerian_path
- [4] Wikipedia, "Marriage Theorem," http://en.wikipedia.org/wiki/Marriage_theorem
- [5] Wikipedia, "Matching," <http://en.wikipedia.org/wiki/Matching>
- [6] Wikipedia, "Stable Marriage Problem," http://en.wikipedia.org/wiki/Stable_marriage_problem